ACCESO A BASES DE DATOS

Acceso BD desde una aplicación:
Java DataBase Connectivity (JDBC)

Acceso Seguro a una BD

http://www.personal.fi.upm.es/~lmengual/bases_datos/bd.html

LUIS MENGUAL GALÁN

Acceso Seguro a una Base de Datos

Acceso Seguro a una BD

Objetivos:

- Conocer las amenazas que puede sufrir la información en entorno de una Base de Datos
- Conocer e implementar servicios y mecanismos de seguridad en Internet
- Gestionar el acceso a una Base de Datos incorporando servicios de seguridad
- Protección de los sistemas de los SGBD en una red corporativa

Índice

- Introducción a la Seguridad en Sistemas Distribuidos
- Modelos de Seguridad en Web:
  - Seguridad en el nivel de Red (IPSec /IPv6)
  - Seguridad en el nivel de transporte (SSL)
  - Seguridad a nivel de aplicación (PGP, S/MIME, SET)
- Redes Internas Corporativas Seguras:
  - Direccionamiento IP privado y traducciones
  - Cortafuegos
  - Shink Hole Routers

Introducción a la Seguridad en Sistemas Distribuidos

Objetivos:

- Conocer las amenazas que puede sufrir la información que se distribuye en una red telemática y el entorno de una Base de Datos
- Analizar los servicios de seguridad disponibles así como los mecanismos asociados
- Presentar los protocolos de seguridad como algoritmo distribuido para implementar mecanismos
- Profundizar en las soluciones prácticas para ofrecer servicios de seguridad basadas en criptografía de clave pública

Bibliografía

- "Network and Internetwork Security Principles and Practice". W. Stallings, Prentice Hall. 1995
- "Criptography and Network Security". 4ª Edition. W. Stallings, Prentice Hall. 2005
- "Comunicaciones y Redes de Computadores", 7ª Edición. W. Stallings, Prentice Hall. 2000
- "Computer Networks", Fourth Edition. A. Tanenbaum, 2002
- "MySQL Administrator´s", S.K. Cabral &K. Murphy, Wiley Publishing, Inc, 2009

Introducción a la Seguridad en Sistemas Distribuidos

Índice:

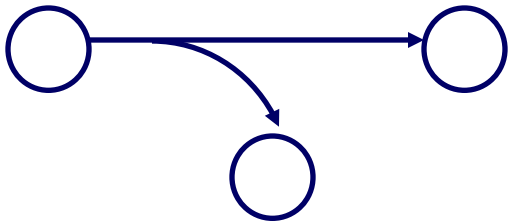
- Amenazas en una Red Telemática
- Servicios de seguridad
- Mecanismos y protocolos de seguridad
- Ejemplos prácticos del uso de la criptografía de clave pública. Uso de los certificados digitales
- Ejemplo criptografía simétrica para distribución de claves

AMENAZAS EN UNA RED DE DATOS (I)

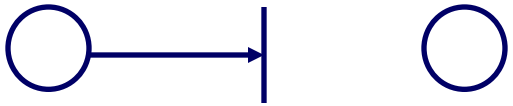
FLUJO NORMAL DATOS

ORIGEN DATOS DESTINO DATOS

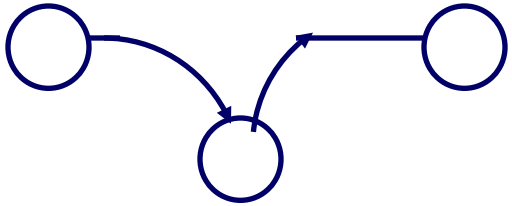
INTERCEPTACIÓN



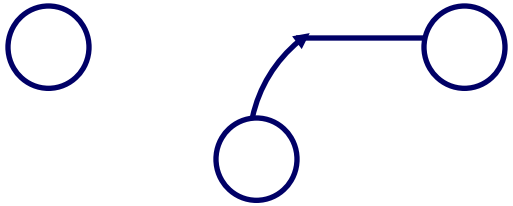
INTERRUPCIÓN



MODIFICACIÓN



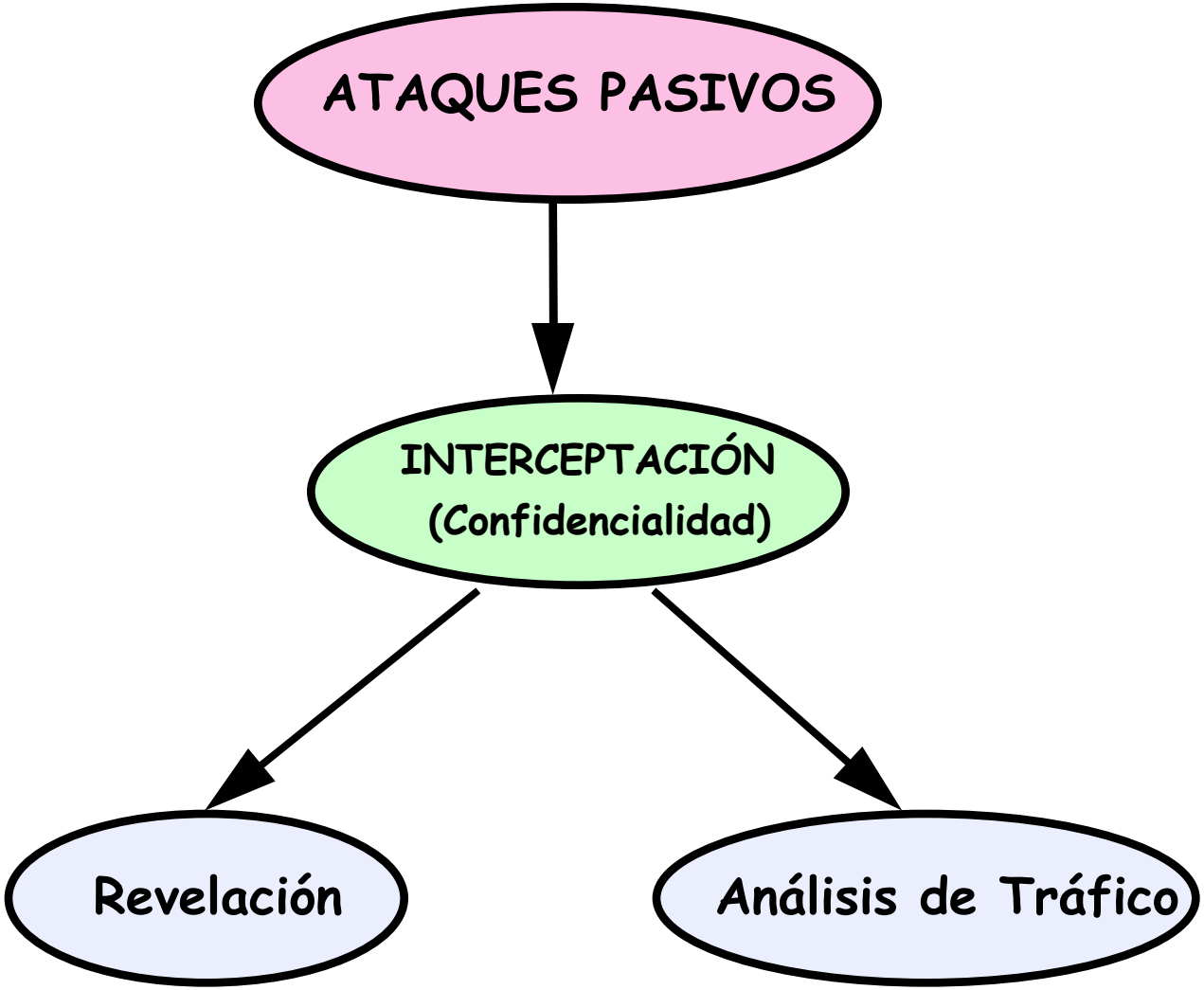
FABRICACIÓN



ATAQUES PASIVOS (I)

- Tienen su origen en la escucha o *monitorización* de una transmisión
- El objetivo es obtener la información que está siendo transmitida
- Son muy difíciles de detectar
- Es posible evitar estos ataques
- Hay que hacer más énfasis en la *prevención* que en la detección.

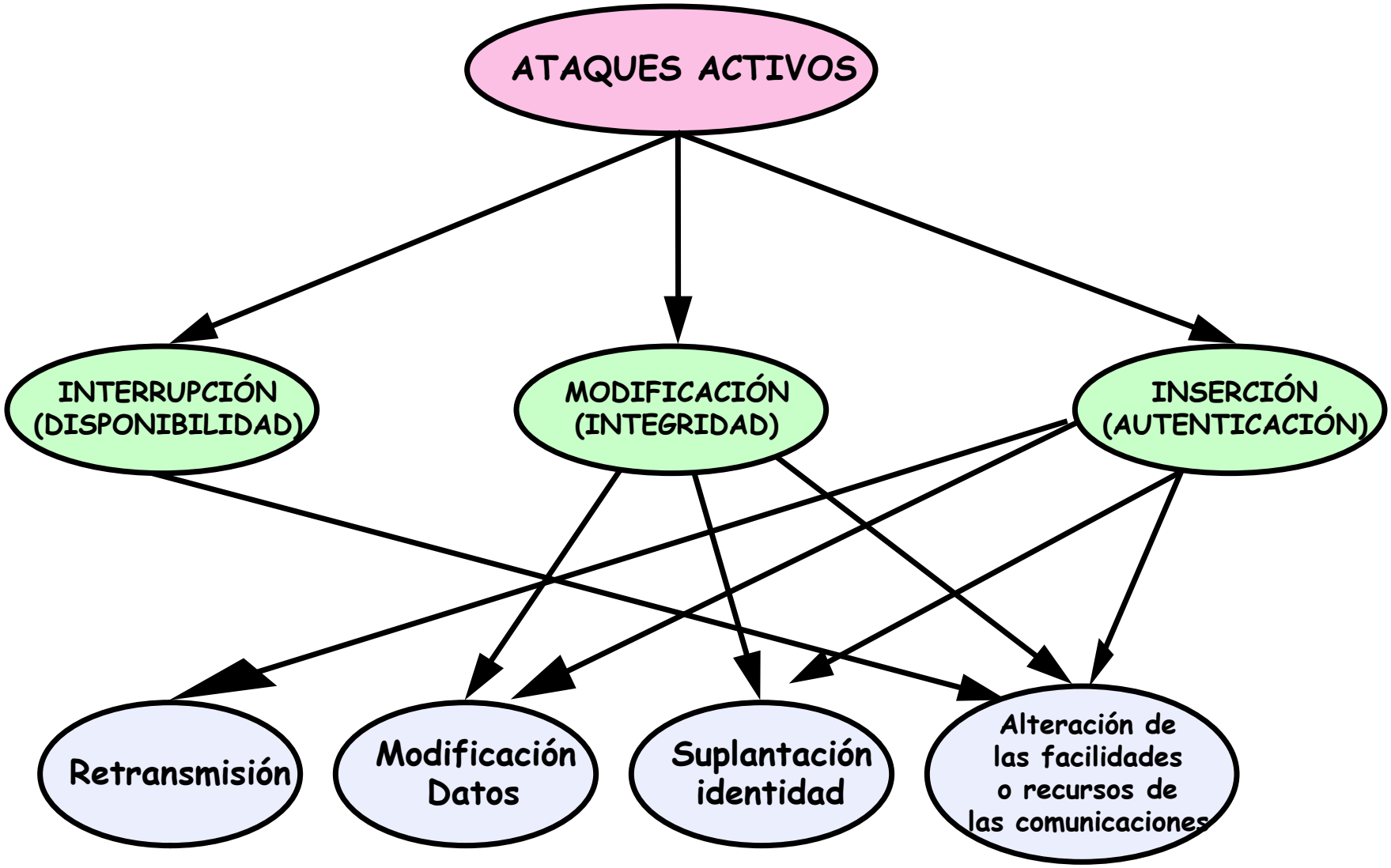
ATAQUES PASIVOS (II)



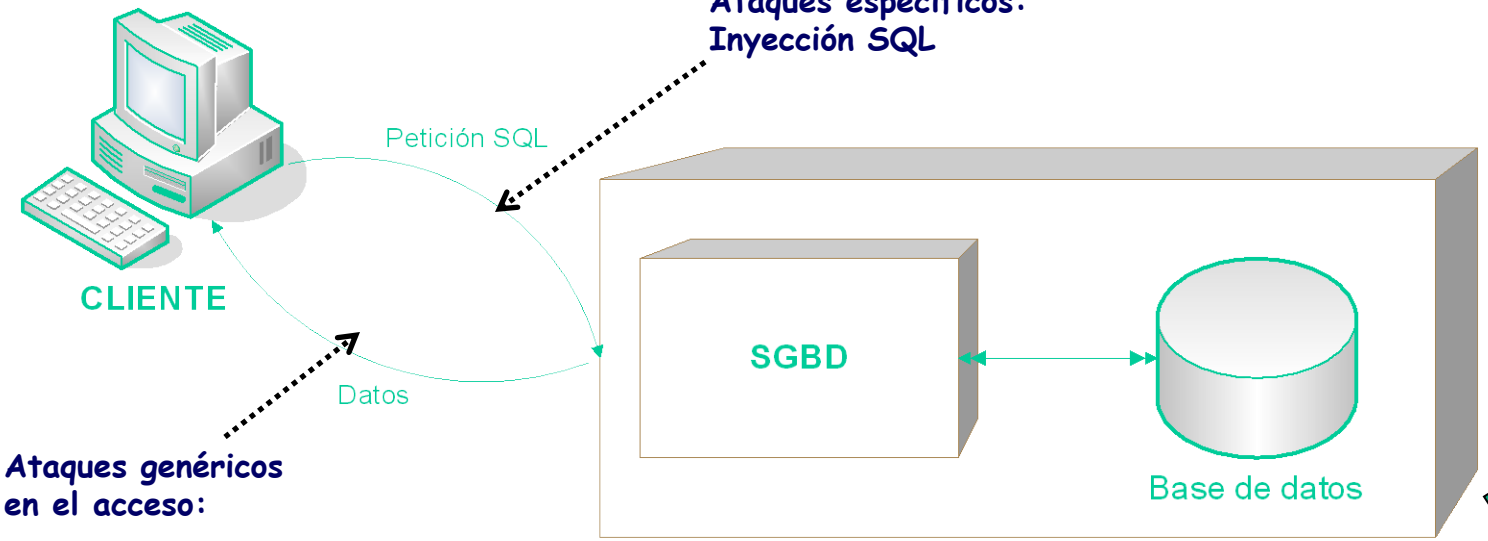
ATAQUES ACTIVOS (I)

- Implican la disposición activa de un intruso:
  - Modificación en el flujo de unidades de datos
  - Creación de unidades de datos fraudulentas
  - Interrupción de las comunicaciones
- Son difíciles de evitar de manera absoluta
- Los servicios de seguridad tratan de detectarlos y recuperarse de cualquier perturbación o retardos ocasionados por ellos.
- Debido a que la detección tiene un efecto disuasivo también puede contribuir a la prevención.

ATAQUES ACTIVOS (II)



ATAQUES A UNA BASE DE DATOS



Ataques genéricos en el acceso:

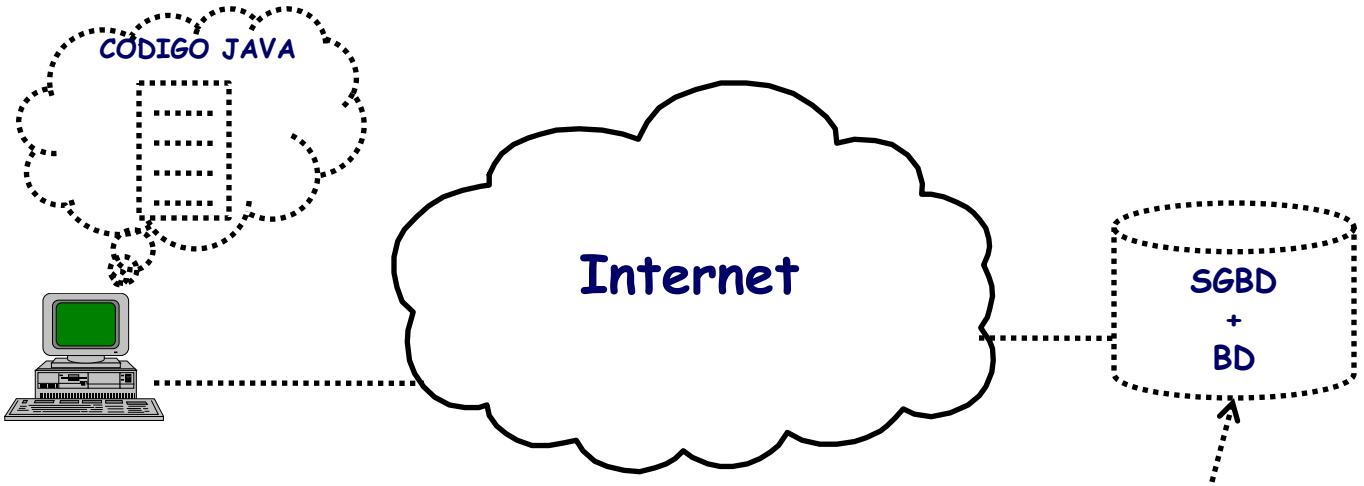
Confidencialidad
Autenticación
Integridad
No repudio

Ataques en el Almacenamiento:

Confidencialidad
Integridad

DNI	Apellidos	Nombre	Puesto	Edad
123456789	Pérez García	Francisco	Director	45
987654321	Martín Gómez	Rául	Ingeniero	30
…	…	…	…	…

ATAQUES A UNA BASE DE DATOS
Acceso a la información (I)



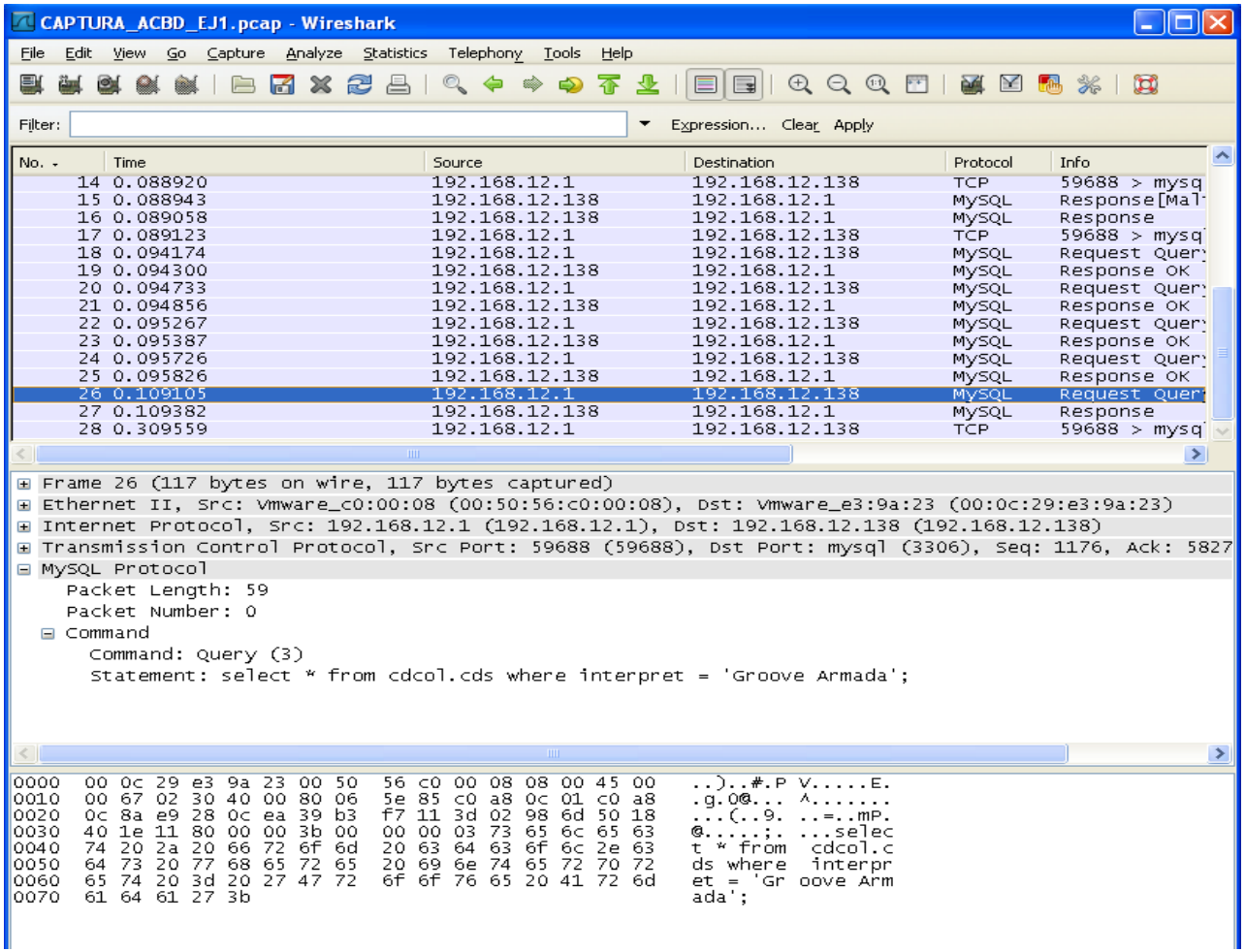
cdcol.cds

titel	interpret	jahr	id
Beauty	Ryuichi Sakamoto	1990	1
Goodbye Country (Hello Nightclub)	Groove Armada	2001	4
Glee	Bran Van 3000	1997	5

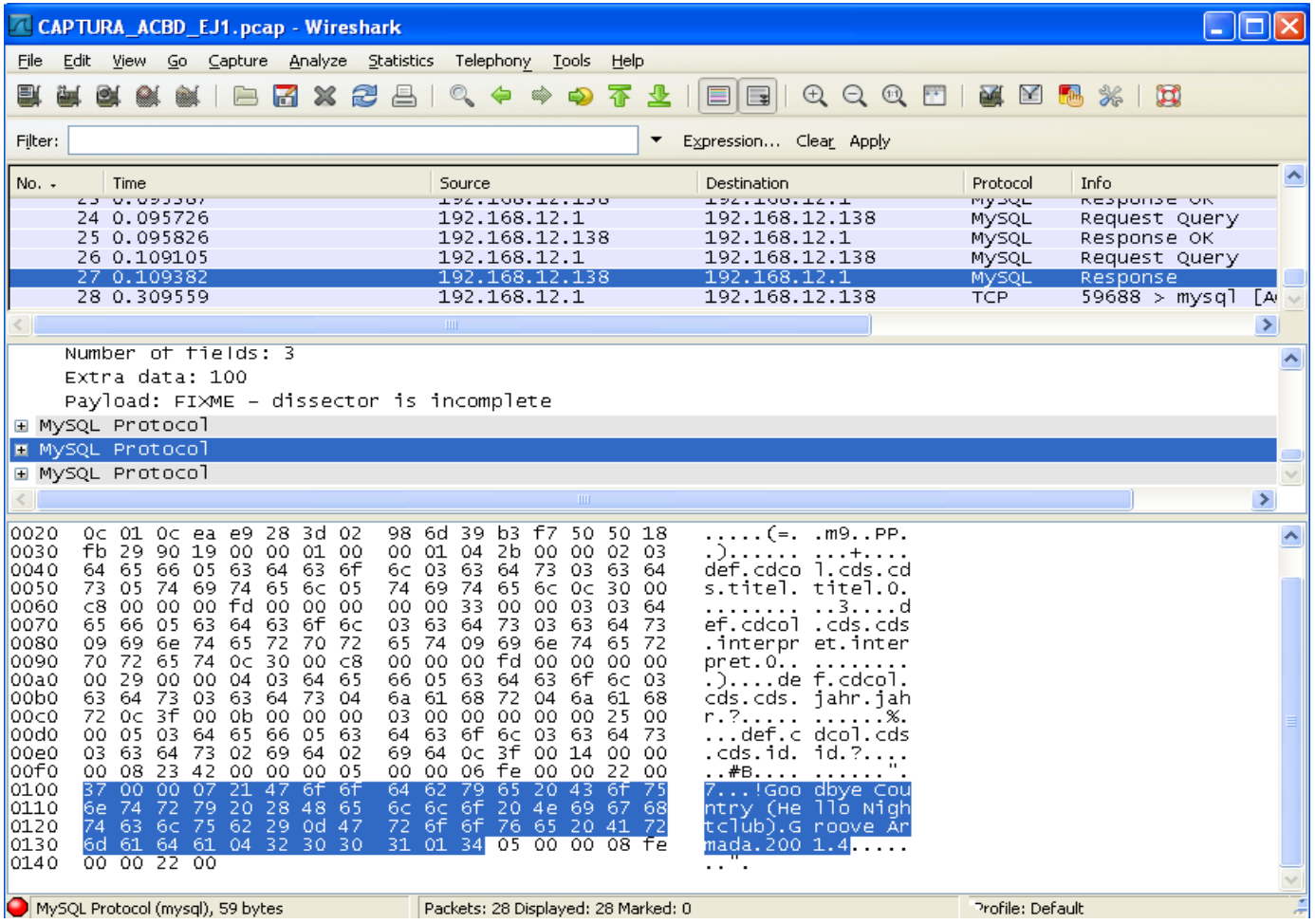
ATAQUES A UNA BASE DE DATOS
Acceso a la información (II)

```
public static void main (String[] args) {
   try
   {
    // Step 1: Load the JDBC driver.
      Class.forName("com.mysql.jdbc.Driver");

    // Step 2: Establish the connection to the database.
    String url = "jdbc:mysql://192.168.12.138:3306/cdcol";
     Connection conn = DriverManager.getConnection(url,"root","");

PreparedStatement ps1;
   ResultSet rs1;
   String resultStr1;

   String query1 = "select * from cdcol.cds where interpret = 'Groove Armada';";
    s1 = conn.createStatement(query1);
    s1 = p1.executeQuery(query1);

while (s1.next()) {
 resultStr1 = rs1.getString("titel")+": "+rs1.getString("interpret")+": "+rs1.getString("jahr")+": "+rs1.getInt("id");
 System.out.println(resultStr1);
 }

   }
   catch (Exception e)
   {     System.err.println("Got an exception! ");
         System.err.println(e.getMessage());
   }
 }
}
```

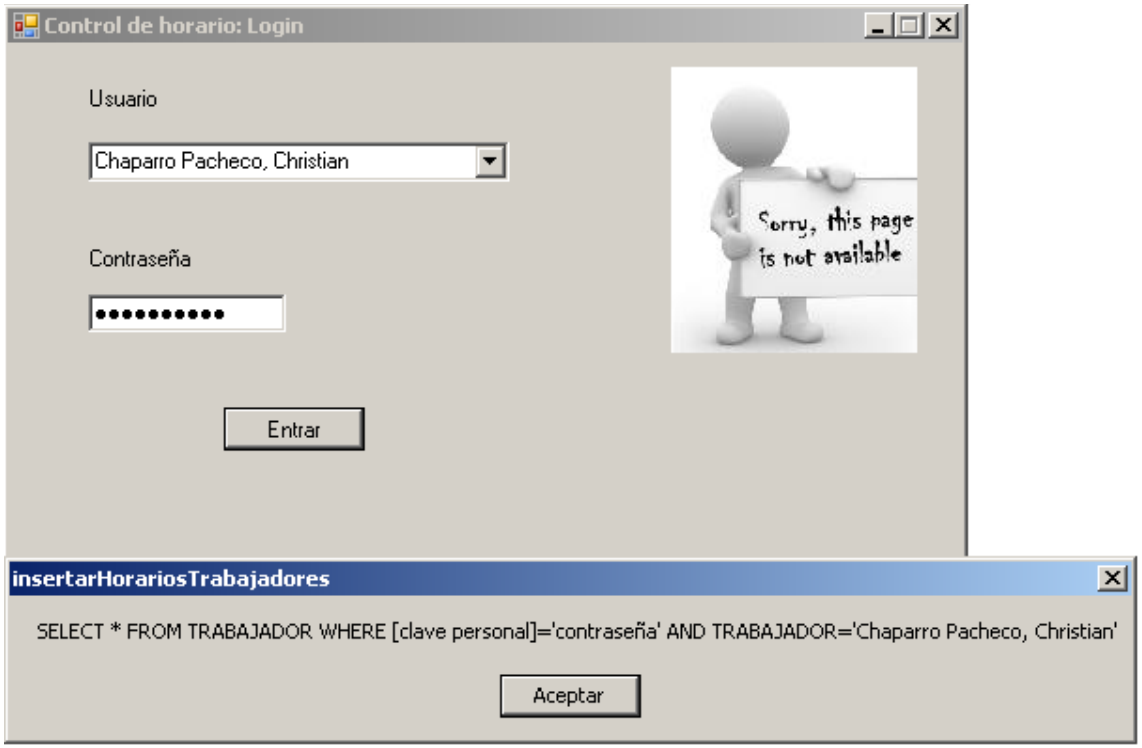
ATAQUES A UNA BASE DE DATOS
Acceso a la información (III)



ATAQUES A UNA BASE DE DATOS
Acceso a la información (IV)



INYECCIÓN SQL (III)



*SELECT * FROM TRABAJADOR WHERE [clave personal]=* **'loqueseteocurra' OR 'a'='a'**
AND TRABAJADOR='" & usuario & "'"

INYECCIÓN SQL (I)

consulta := "SELECT * FROM usuarios WHERE nombre = '" + nombreUsuario + "';"

SELECT * FROM usuarios WHERE nombre = 'Alicia';

Pero si un operador malintencionado escribe como nombre de usuario a consultar:

"Alicia'; DROP TABLE usuarios; SELECT * FROM datos WHERE nombre LIKE '% "

se generaría la siguiente consulta SQL,

(el color verde es lo que pretende el programador, el azul es el dato,
y el rojo, el código SQL inyectado):

```
SELECT * FROM usuarios WHERE nombre = 'Alicia';
DROP TABLE usuarios;
SELECT * FROM datos WHERE nombre LIKE '%';
```

INYECCIÓN SQL (II)

En lenguaje Java, se puede usar la clase PreparedStatement

En lugar de:

```
Connection con = (acquire Connection)
 Statement stmt = con.createStatement();
ResultSet rset = stmt.executeQuery("SELECT * FROM usuarios WHERE nombre = '" + nombreUsuario + "';");
```

se puede usar parametrización o escape de variables, como se indica en los siguiente apartados.

Parametrización de sentencias SQL

```
Connection con = (acquire Connection)
PreparedStatement pstmt = con.prepareStatement("SELECT * FROM usuarios WHERE nombre = ?");
pstmt.setString(1, nombreUsuario);
ResultSet rset = pstmt.executeQuery();
```

SERVICIOS DE SEGURIDAD (I)
X.800 / ISO 7498-2

- **CONFIDENCIALIDAD**
  - Protección de los datos frente a intrusos
  - Variantes:
    - Orientada a conexión
    - No orientada a Conexión
    - Selectiva
    - Aplicada al análisis de tráfico
- **AUTENTICACIÓN**
  - Garantía del origen de los datos y de las entidades implicadas
  - Variantes:
    - En comunicaciones no orientadas a conexión
    - En comunicaciones orientadas a conexión

SERVICIOS DE SEGURIDAD (II)
X.800 / ISO 7498-2

- **INTEGRIDAD**
  - Garantía de la no alteración de la información
  - Variantes:
    - En comunicaciones orientadas a conexión (con y sin mecanismos de recuperación, aplicada a campos selectivos)
    - En comunicaciones no orientadas a conexión (aplicadas o no a campos selectivo)
- **NO REPUDIO**
  - Evita que tanto el emisor como el receptor nieguen haber transmitido un mensaje
  - Variantes:
    - Con prueba de origen
    - Con prueba de entrega

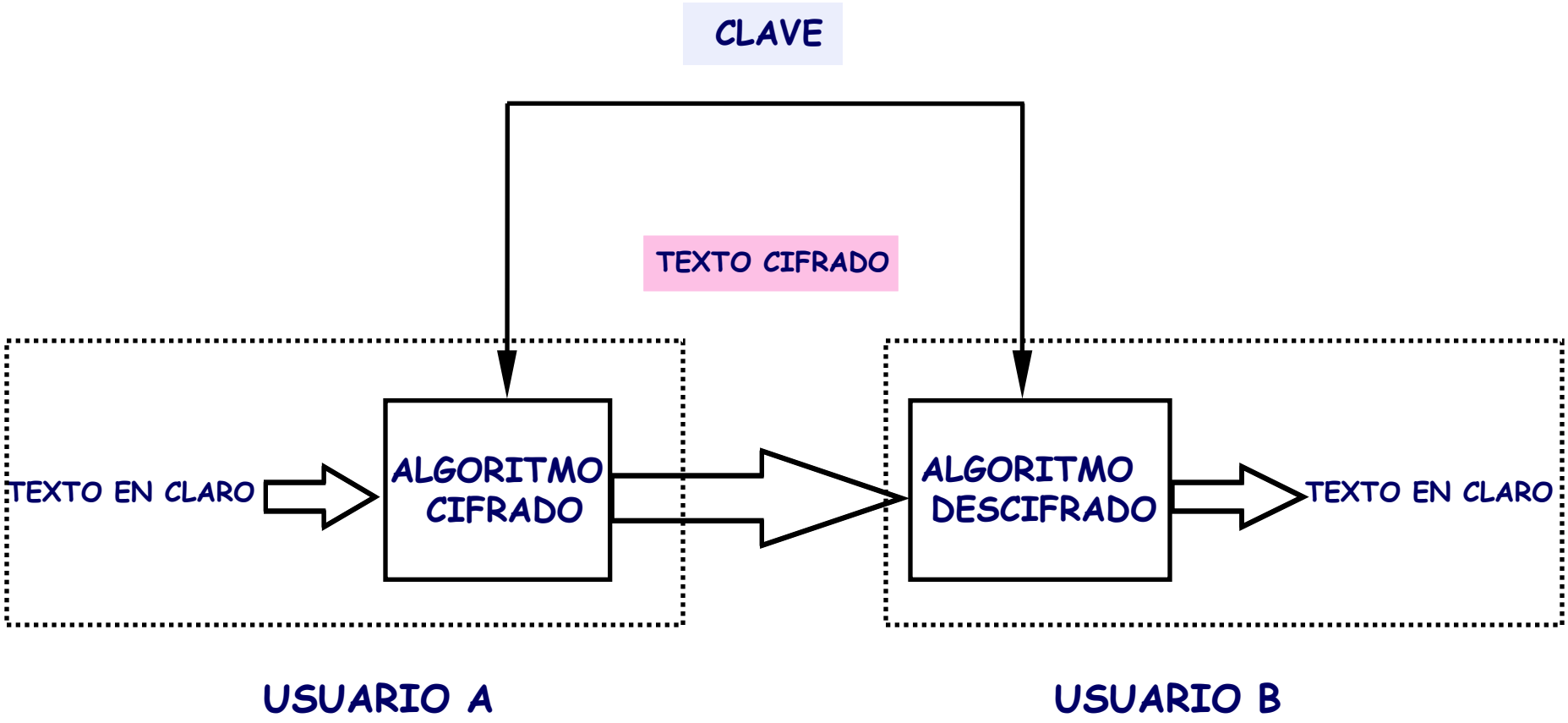
SERVICIOS DE SEGURIDAD (III)
X.800 / ISO 7498-2

- CONTROL DE ACCESO
  - Capacidad para permitir o denegar el uso de un objeto por un sujeto
  - Debe de proporcionar tres elementos esenciales:
    - Autenticación (Authentication)
      - Quién puede acceder
    - Autorización (Authorization)
      - Qué puede hacer un usuario
    - Trazabilidad (Accountability)
      - Identificar qué ha hecho un usuario

MECANISMOS Y PROTOCOLOS

- Mecanismos de seguridad:
  - Cifrado simétrico y asimétrico
  - Intercambio de Autenticación
    - Funciones Hash. Códigos de autenticación de Mensajes.
    - Actualidad de los mensajes
  - Firma digital
- Protocolos de seguridad:
  - Proporcionan autenticación (Mecanismos de Desafío/respuesta, Sellos de tiempo)
  - Distribuyen claves de sesión

CIFRADO SIMÉTRICO (I)



CIFRADO SIMÉTRICO (II)

- Aspectos a considerar para la elección del algoritmo:
  - Longitud Clave
  - Robustez algoritmo
  - Velocidad de cifrado

ALGORITMOS DE CIFRADO SIMÉTRICO (I)

- DES (Data Encryption Standard)
  - Adoptado en 1977 por el NIST (National Institute of Standars and Technology)
  - Los datos a cifrar se procesan en bloques de 64 bits
  - La longitud de la clave es de 56 bits
- Triple DES
  - Tres ejecuciones del algoritmo DES
  - Longitud de clave efectiva de 168 bits
- IDEA (International Data Encyption Algorithm)
  - Desarrollado por Xuejia Lai y J. Massey (Swiss Federal Institute of Technology)
  - Los datos a cifrar se procesan en bloques de 64 bits
  - La longitud de la clave es de 128

CIFRADO SIMÉTRICO
ALGORITMOS DE CIFRADO (II)

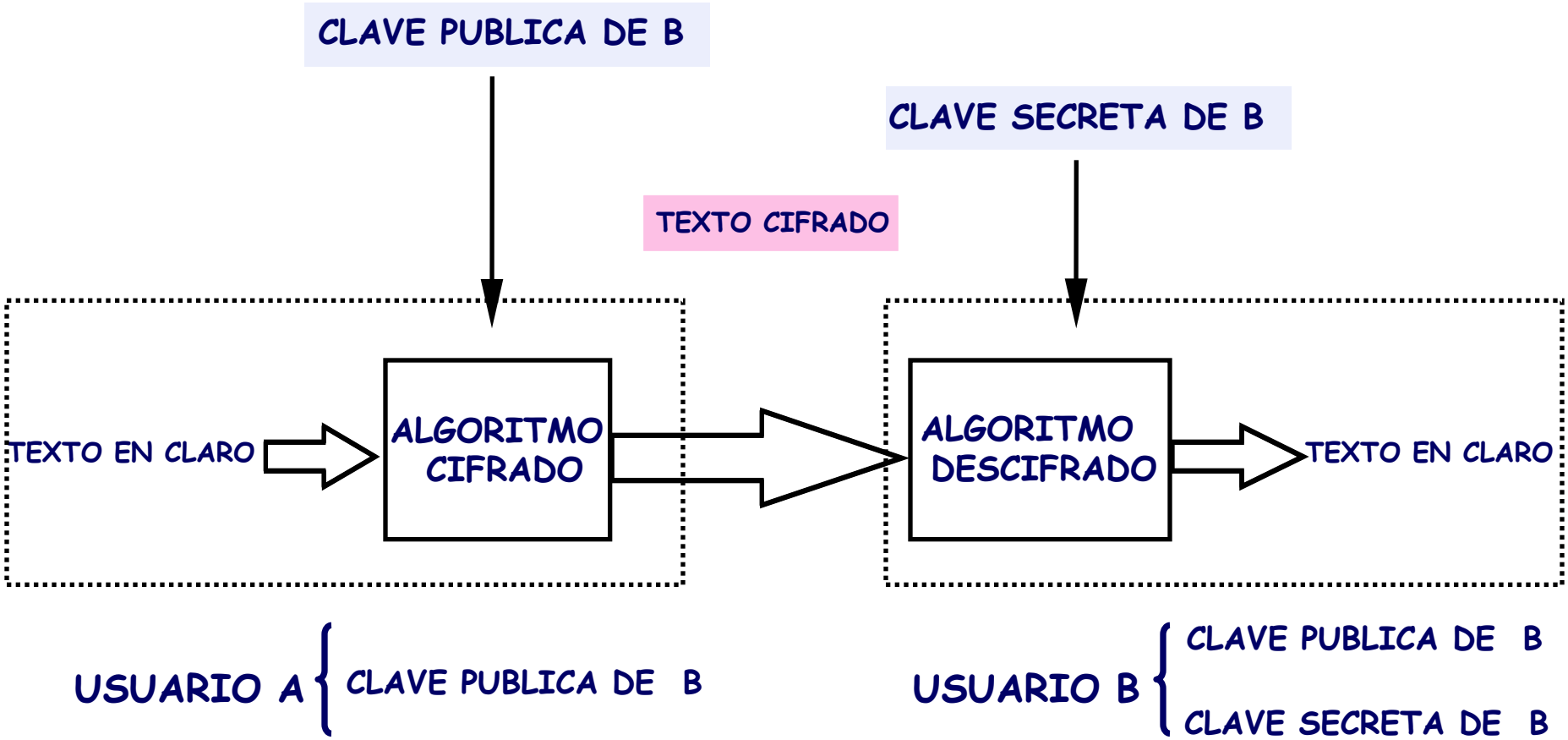
- **RC5**
  - Desarrollado por Ron Rivest
  - Los datos a cifrar se procesan en bloques de longitud 32, 64 o 128 bits
  - Longitud de clave variable (0 a 1024Bits)
- **Blowfish**
  - Desarrollado por Bruce Schneier
  - Los datos a cifrar se procesan en bloques de 64 bits
  - La longitud de la clave es de hasta 448 bits
- **Rijndael**
  - Desarrollado V. Rijmen J. Daemen, y propuesto como AES (Advanced Encryption Standard) por el NIST
  - Los datos a cifrar se procesan en bloques de longitud variable: 128, 192 ó 256 Bits
  - Longitud de clave variable: 128, 192 ó 256 Bits

CIFRADO SIMÉTRICO
ALGORITMOS DE CIFRADO (III)

- Velocidades aproximadas de cifrado (Pentium 400Mhz)*
  - Blowfish: 22Mbytes/s
  - RC5: 17 Mbytes/s
  - DES: 8,8 Mbytes/s
  - IDEA: 8 Mbytes/s
  - Triple DES: 3,7 Mbytes/s

*Referencia Blowfish Website

CIFRADO ASIMÉTRICO (I)
(Ó DE CLAVE PÚBLICA)



CIFRADO ASIMÉTRICO (II)
(Ó DE CLAVE PÚBLICA)

- Se generan un par de claves complementarias (pública y secreta)
- La clave secreta (privada) no sale del sistema del usuario
- La clave pública está disponible al resto de los usuarios del sistema
- Las claves son reversibles
- No debe ser factible obtener la clave secreta a partir de la pública y viceversa

ALGORITMOS DE CIFRADO ASIMÉTRICO

- Diffie-Hellman
  - Primer sistema de Criptografía de clave pública. 1975
- RSA
  - Desarrollado por Rivest-Shamir-Adelman en el MIT en 1978
  - Es el algoritmo más utilizado en criptografía asimétrica
- Elliptic curve Diffie-Hellman (ECDH)
  - Claves más cortas
  - Más rápido que RSA

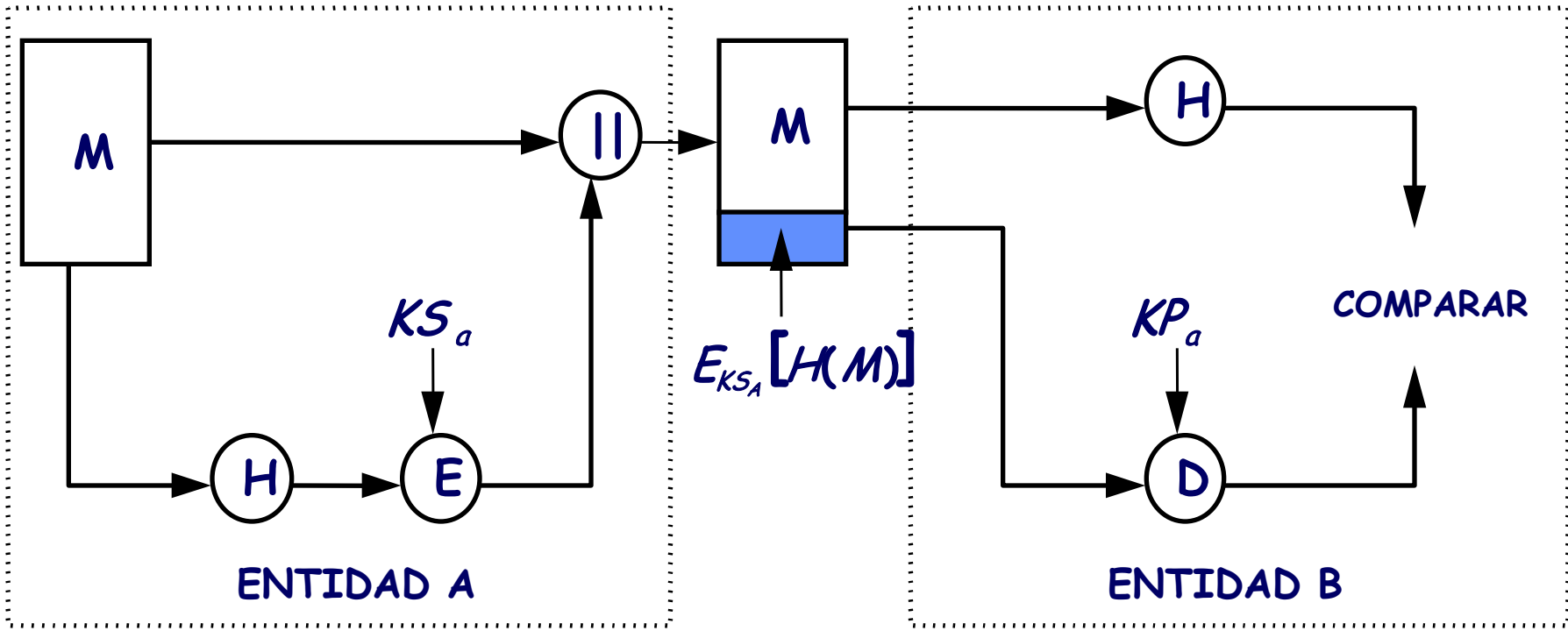
FUNCIÓN HASH (I)
(RESÚMEN)

- Es una función que aplicada a un bloque de longitud arbitraria produce una salida de longitud fija
- Propiedades de una función Hash
  - Para un *h* dado deber ser computacionalmente imposible encontrar un *m* tal que *h=H(m)*
  - Para un *m* dado deber ser computacionalmente imposible encontrar un *m´* tal que *H(m´)=H(m)*
  - No debe de ser factible computacionalmente encontrar un par (m,m´) tal que *H(m)=H(m´)*

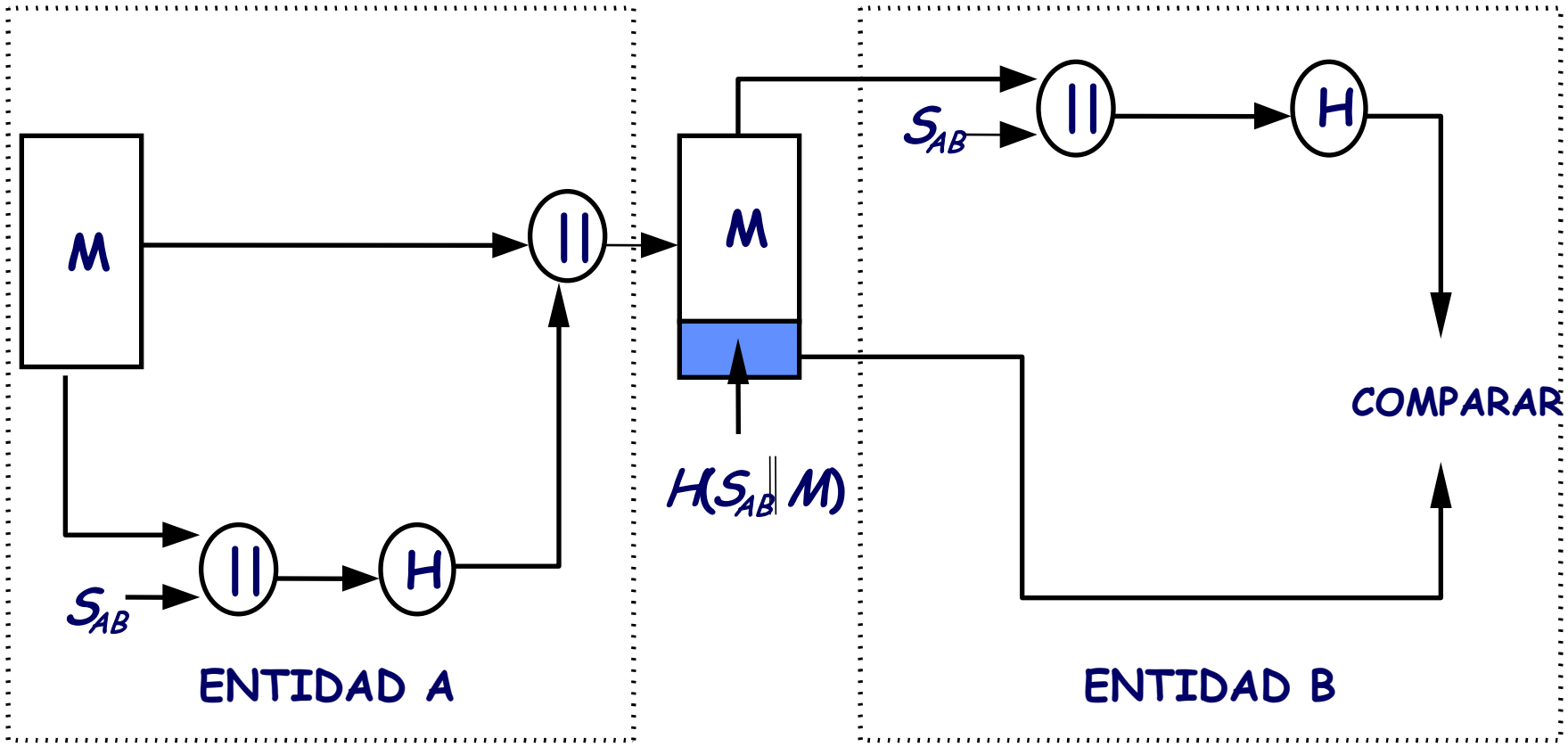
ALGORITMOS HASH

- MD5 (Message Digest Algorithm)
  - Desarrollado por Ron Rivest en el MIT (RFC 1321)
  - Se toma una entrada de longitud arbitraria y se genera un resumen de 128 bits
  - La entrada se procesa en bloques de 512 bits
- SHA (Secure Hash Algorithm)
  - Desarrollado por NITS (FIPS PUB 180)
  - Se toma una entrada de longitud arbitraria y se genera un resumen de 160 bits
  - La entrada se procesa en bloques de 512 bits
- SHA 2 (Secure Hash Algorithm)
  - SHA2 mejora la seguridad de SHA1 (FIPS PUB 180-2)
  - Longitudes salida 256 bits
  - La entrada se procesa en bloques de 256/512 bits
- SHA 3, Keccak. NIST FIPS 202 the "SHA-3" Standard
  - Nuevo algoritmo hash. Ganador del NIST hash Funtion Competition. 2012
  - Longitudes salida 512 bits
  - La entrada se procesa en bloques de 1600 bits

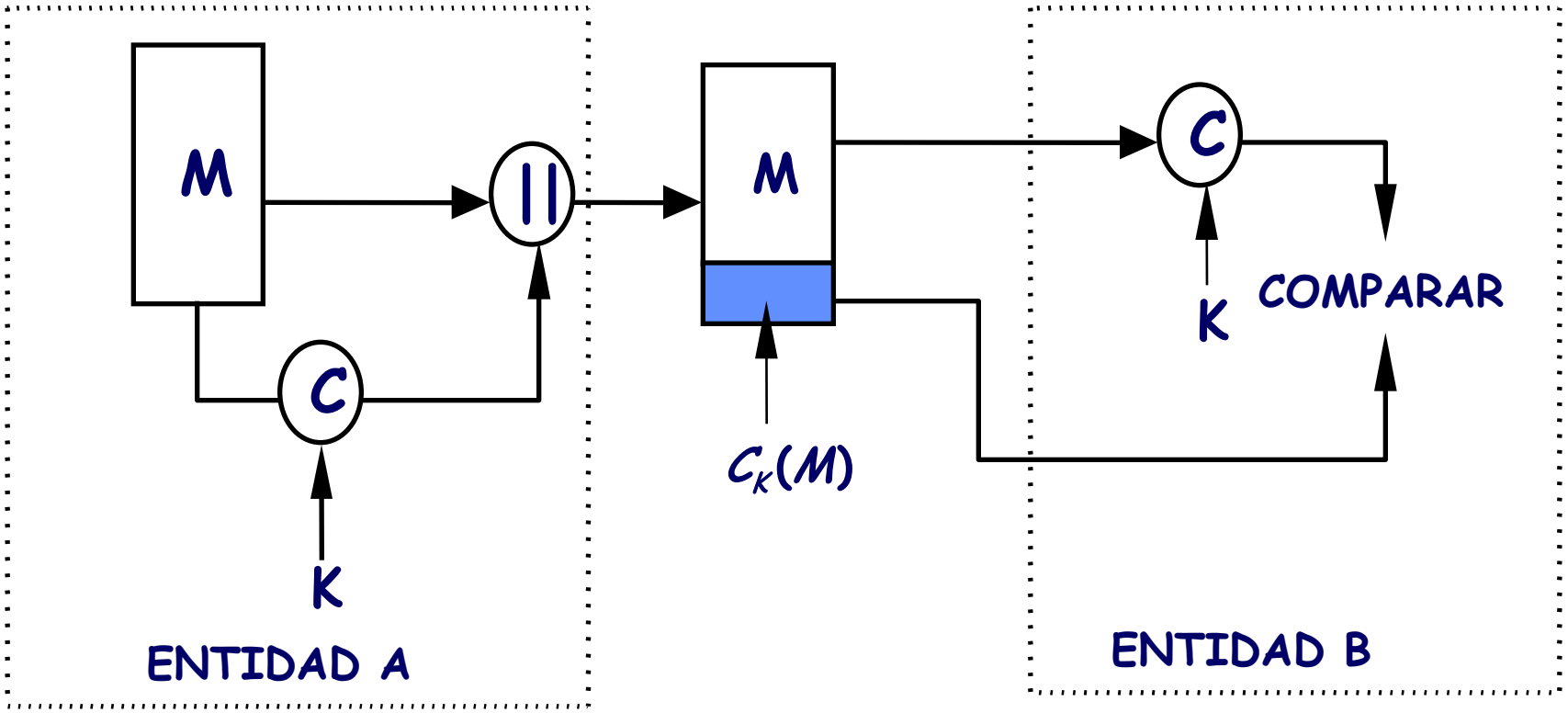
FUNCIÓN HASH (II)



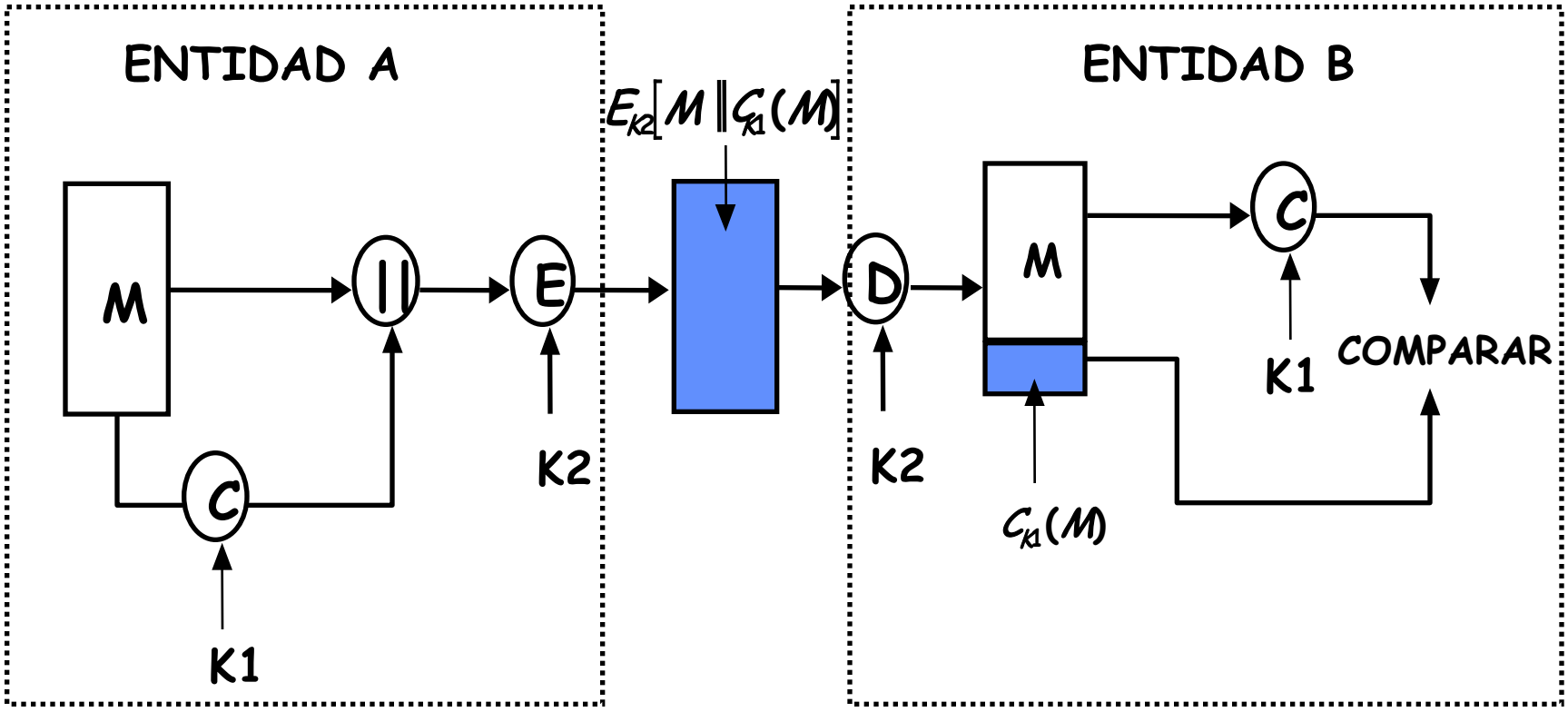
FUNCIÓN HASH (III)



CÓDIGO DE AUTENTICACIÓN DE MENSAJES (I)



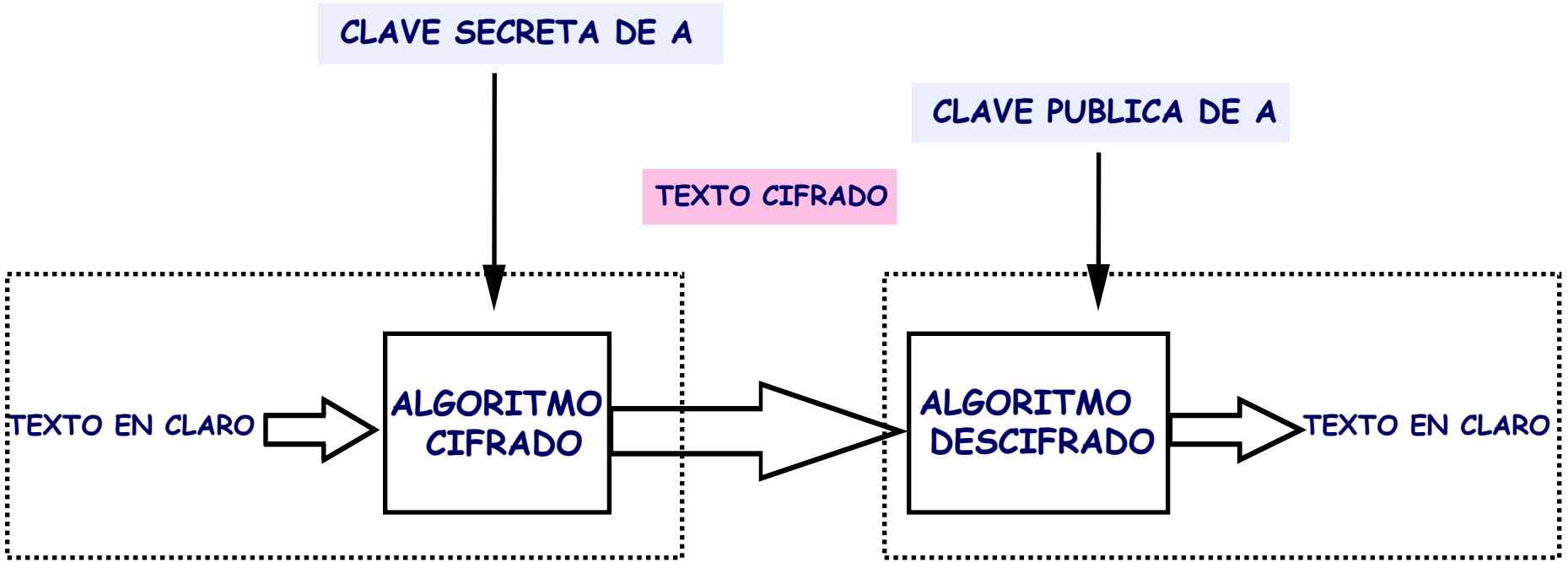
CÓDIGO DE AUTENTICACIÓN DE MENSAJES (II)



FIRMA DIGITAL (I)

- Debe ser posible verificar al autor y el tiempo de la firma
- Debe ser posible autentificar los contenidos de los mensajes en el tiempo de la firma
- La firma debe estar disponible por las entidades para resolver disputas

FIRMA DIGITAL (II)



USUARIO A { CLAVE PUBLICA DE A ; CLAVE SECRETA DE A }

USUARIO B { CLAVE PUBLICA DE A }

EJEMPLOS PRÁCTICOS DE LA UTILIZACIÓN DE LA CRIPTOGRAFÍA DE CLAVE PÚBLICA.

USO DE LOS CERTIFICADOS DE CLAVE PÚBLICA

UTILIZACIÓN DE LA CRIPTOGRAFÍA DE CLAVE PÚBLICA PARA AUTENTICACIÓN

- La entidad A quiere autenticar a B
- La entidad B tiene un par de claves (una pública y otra privada)
- La entidad A conoce la clave pública de B

A:->B: mensaje_aleatorio

B:->A:cifar(KPrivadaB, mensaje_aleatorio);

UTILIZACIÓN DE RESÚMENES DE MENSAJES

- La función Hash (resumen) toma un conjunto de datos de entrada y genera un resumen de longitud fija
- La función Hash es difícilmente reversible
- Se puede usar para proporcionar autenticación, integridad y firma digital

A:->B: msj_aleatorio

B:->A: msj_aleatorio, cifar(KPrB, H(msj_aleatorio));

FIRMA DIGITAL

MANEJO DE CLAVES PÚBLICAS (I)
¿Por qué son necesarios los certificados?

A:->B: "Hola"
B:->A:"Hola, Yo soy B", KPuB<clave pública de B>
A:->B: "Pruébala"
B:->A:"Soy B", cifrar(KPrB, H["Soy B"]);

- **Cualquier entidad puede suplantar a B**
- **Basta únicamente tener una clave pública y otra privada**

¿QUÉ ES UN CERTIFICADO?

- Un certificado digital (ó certificado de clave pública) establece la identidad de un usuario en un red
- Es equivalente a una tarjeta de crédito o a un carnet de conducir
- La estructura de un certificado está definida en el estándar ITU X.509
- En una red:
  - Los servidores pueden ser configurados para permitir el acceso a usuarios con ciertos certificados
  - los clientes pueden ser configurados para confiar en servidores que presentan ciertos certificados.

¿QUÉ ES UN CERTIFICADO?

```
Certificate:
    Data:
        Version: v3 (0x2)
        Serial Number: 3 (0x3)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer: OU=Ace Certificate Authority, O=Ace Industry, C=US
        Validity:
            Not Before: Fri Oct 17 18:36:25 1997
            Not  After: Sun Oct 17 18:36:25 1999
        Subject: CN=Jane Doe, OU=Finance, O=Ace Industry, C=US
        Subject Public Key Info:
            Algorithm: PKCS #1 RSA Encryption
            Public Key:
                Modulus:
                    00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86:
                    ed:27:40:4d:86:b3:05:c0:01:bb:50:15:c9:de:dc:85:19:22:
                    43:7d:45:6d:71:4e:17:3d:f0:36:4b:5b:7f:a8:51:a3:a1:00:
                    98:ce:7f:47:50:2c:93:36:7c:01:6e:cb:89:06:41:72:b5:e9:
                    73:49:38:76:ef:b6:8f:ac:49:bb:63:0f:9b:ff:16:2a:e3:0e:
                    9d:3b:af:ce:9a:3e:48:65:de:96:61:d5:0a:11:2a:a2:80:b0:
                    7d:d8:99:cb:0c:99:34:c9:ab:25:06:a8:31:ad:8c:4b:aa:54:
                    91:f4:15
                Public Exponent: 65537 (0x10001)
        Extensions:
                                .....

    Signature:
        Algorithm: PKCS #1 MD5 With RSA Encryption
        Signature:
            6d:23:af:f3:d3:b6:7a:df:90:df:cd:7e:18:6c:01:69:8e:54:65:fc:06:
            30:43:34:d1:63:1f:06:7d:c3:40:a8:2a:82:c1:a4:83:2a:fb:2e:8f:fb:
            f0:6d:ff:75:a3:78:f7:52:47:46:62:97:1d:d9:c6:11:0a:02:a2:e0:cc:
            2a:75:6c:8b:b6:9b:87:00:7d:7c:84:76:79:ba:f8:b4:d2:62:58:c3:c5:
            b6:c1:43:ac:63:44:42:fd:af:c8:0f:2f:38:85:6d:d6:59:e8:41:42:a5:
            4a:e5:26:38:ff:32:78:a1:38:f1:ed:dc:0d:31:d1:b0:6d:67:e9:46:a8:
             dd:c4
```

NÚMERO DE SERIE

AUTORIDAD DE CERTIFICACIÓN

PERÍODO DE VALIDEZ

PROPIETARIO DEL CERTIFICADO

CLAVE PÚBLICA DEL PROPIETARIO

FIRMA DE LA AUTORIDAD DE CERTIFICACIÓN

¿Cómo un cliente autentifica la identidad de un servidor?

¿Esta la fecha de hoy dentro del período de validez del certificado?

¿Es fiable la Autoridad de Certificación que firma el certificado?

¿Es válida la firma del certificado?

¿Cómo un servidor autentifica la identidad de un cliente ?

NÚMERO DE SERIE:

.....

AUTORIDAD DE CERTIFICACIÓN:

.....

PERÍODO DE VALIDEZ:

.....

PROPIETARIO DEL CERTIFICADO: CLIENTE

.....

CLAVE PÚBLICA DEL PROPIETARIO:

.....

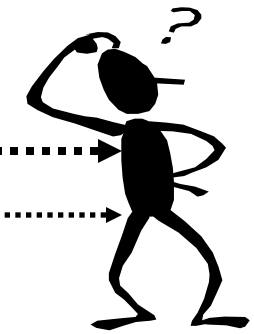
FIRMA DE LA AUTORIDAD DE CERTIFICACIÓN:

¿Esta la fecha de hoy dentro del período de validez del certificado?

¿Es fiable la Autoridad de Certificación que firma el certificado?

¿Es válida la firma del certificado?

¿La clave pública del cliente valida su firma digital?



DATOS CLIENTE
FIRMA DIGITAL CLIENTE

MANEJO DE CLAVES PÚBLICAS (II)
¿Por qué son necesarios los certificados?

A:->B: "Hola"
B:->A:"Hola, Yo soy B", <Certificado de B>
A:->B: "Pruébala"
B:->A:"Soy B", cifrar(KPrB, H["Soy B"]);

- Un intruso no podría crear una firma digital cifrando con la clave privada de B

A:->M: "Hola"
M:->A:"Hola, Yo soy B", <certificado de B>
A:->M: "Pruebalo"
M:->A: ?????????

INTERCAMBIO DE CLAVES DE SESIÓN CON CRIPTOGRAFÍA DE CLAVE PÚBLICA

A:->B:encrypt(KPuB, clave_sesión);

A:->B: "Hola"
B:->A:" Yo soy B", <certificado de B>
A:->B: "Pruébalo"
B:->A:" Soy B", cifrar(KPrB, H["Soy B"]);
A:->B: "ok,Aquí esta la clave", cifrar(KPuB,<clave sesión>);
B:->A:cifrar(clave sesión, <algunos mensajes>);

- **Un intruso podría suplantar a B y alterar los mensajes cifrados**

CÓDIGO DE AUTENTICACIÓN DE MENSAJES

- El mensaje lleva adicionalmente un Código de Autenticación de Mensajes (MAC)
- Las entidades pares comparten una clave secreta

MAC:=H[mensaje, clave autenticación]

A:->B: "Hola"
B:->A:" Yo soy B", <certificado de B>
A:->B: "Pruébalo"
B:->A:" Soy B", cifrar(KPrB, H["Soy B"]);
A:->B: "ok,Aquí están las claves", cifrar(KPuB,<clave_ sesión +clave_autenticación>);
B:->A:cifrar(clave sesión, <algunos mensajes, MAC>);